H2020 European Union funding
for Research & Innovation

Measurement and Architecture for a Middleboxed Internet

H2020-ICT-688421

Data Management Plan

Author(s):	ZHAW	Stephan Neuhaus (ed.)
	ETH	Mirja Kühlewind
	ETH	Brian Trammell
	ULg	Benoit Donnet
	ZHAW	Roman Müntener

Document Number:	D4.1
Internal Reviewer:	Diego R. Lopez
Due Date of Delivery:	30 April 2016
Actual Date of Delivery:	31 June 2016
Dissemination Level:	Public

Disclaimer

The information, documentation and figures available in this deliverable are written by the Measurement and Archticture for a Middleboxed Internet (MAMI) consortium partners under EC co-financing (project H2020-ICT-688421) and does not necessarily reflect the view of the European Commission.

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The user uses the information at its sole risk and liability.

Contents

Disclaimer 2

Executive Summary 4

1 Introduction 5

2 MAMI Observatory Architecture and Implementation 6
2.1 Software Management 7

3 Data Management Plan 8
3.1 Data Set Description 8
3.2 Standards and Metadata 9
3.3 Data Sharing 9
3.4 Archiving and Preservation (Incl. Storage and Backup) 10

4 Data Sources 11
4.1 PathSpider 11
4.2 Tracebox 12
4.3 Copycat 12
4.4 Revelio 14

Executive Summary

Horizon 2020 projects can participate in a limited and flexible pilot action on open access to research data. Participating projects must develop a Data Management Plan (DMP) specifying which data will be openly accessible. This deliverable contains that data management plan.

As per the *Guidelines on Data Management in Horizon 2020* (Version 2.1, 15 February 2016)[1], this document covers:

- The handling of research data during & after the project (Sections 2 and 3)
- What data will be collected, processed or generated (see Section 4)
- What methodology & standards will be applied (Section 2)
- Whether data will be shared/made open access & how (Sections 2 and 3.3)
- How data will be curated & preserved (Section 3.4)

[1] http://ec.europa.eu/research/participants/data/ref/h2020/grants_manual/hi/oa_pilot/h2020-hi-oa-data-mgt_en.pdf

1 Introduction

One of the primary objectives of the MAMI project is the collection and investigation of data about middlebox manipulation of packets on the Internet. To this end, MAMI stores data in a *Middlebox Observatory* (or 'Observatory' for short). This Observatory is meant primarily for data generated in the MAMI project itself, but may store or cache data from other measurement projects as well.

Data being offered by the MAMI data Observatory falls into three categories.

Data Generated by MAMI. These are data generated by MAMI's own probes and experiments. For these data sets, MAMI Observatory is the authoritative source. One example of such data is Pathspider data (see Section 4.1).

Data Held by MAMI. These are data that were not generated by MAMI, but which are held within the MAMI Observatory, for example because they are otherwise not available on demand. For these data, MAMI can also be viewed as an authoritative source.

Data Cached by MAMI. These are data that are not generated by MAMI, which may be available online, for which MAMI is not the authoritative source, and which are held by the MAMI Observatory only for convenience.

At the time of writing, it is not at all clear that there will be any data of the second category (held authoritatively by MAMI, but not generated by it). We include this category here for completeness' sake in case data of that category should indeed exist. It also not certain that data of the third category (data cached by MAMI) will be used in MAMI except as interesting background data, e.g., to calibrate MAMI's own measurements.

Since MAMI data contains IP addresses and other potentially Personally Identifiable Information (PII), MAMI can not give out these raw data sets to everyone (see Section 3.3). Instead, MAMI runs a query system on top of the Observatory that will aggregate MAMI data to so called *Observations* and thereby only return data that is free of PII (Section 2). Researchers can however ask for an agreement with the MAMI consortium that would allow them access to the raw data files.

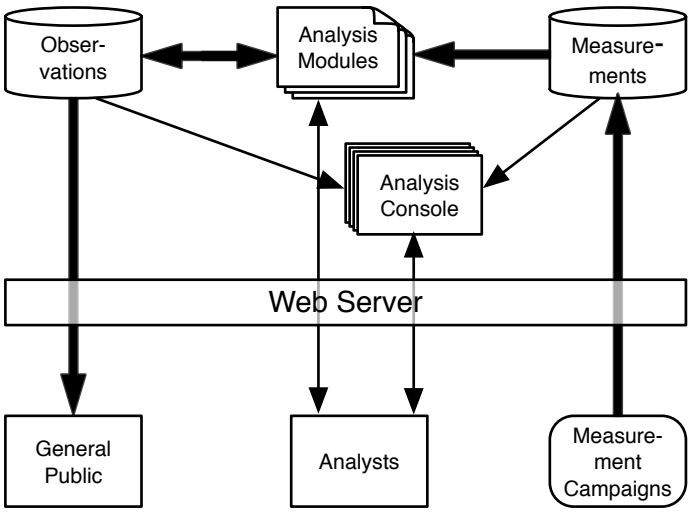

Figure 1: Management Architecture

2 MAMI Observatory Architecture and Implementation

The management architecture is depicted in Figure 1. All access to the data management infrastructure is mediated through a web server, nginx in this case. The data itself is stored in two databases. One stores observations in the format specified by MAMI project, and is used to drive the publicly-accessible observatory website, as well as exploratory analysis. The other database holds the raw data that from which these observations are derived. This data might be available in different formats depending on the tool(s) and methodologies used to collect the data (see section 4). Each data set might have its own analysis module to derive observations. The raw measurement data are stored in a Hadoop filesystem (HDFS) node. Observations are stored in a MongoDB NoSQL database.

Access to these data falls into three categories:

- **Measurement Campaigns.** These may be people or machines. They are equipped with tokens that allow them to upload raw measurements to the measurement infrastructure.
- **Analysts.** These are people who write analysis modules that turn measurements and other observations into more observations for exploratory analysis of both measurements and observations.
- **General Public.** Members of the general public can access observations based on a web frontend.

The analysis modules will be written in a variety of languages, with a preference for Python. Whenever new measurements arrive in the measurement database, appropriate analysis modules will be triggered that transform measurements into observations which can even trigger other analysis modules that derive observation based on other observations.

The analysis consoles are implemented using Jupyterhub, a multi-user Jupyter server which enables the use of Python facilities for data analysis such as Pandas[1].

The MAMI Observatory is managed by ZHAW; Dr. Stephan Neuhaus is the main contact.

2.1 Software Management

We use a software stack that consists exclusively of open-source software for the storage and evaluation of measurements and observations. The list of this software contains Linux (Ubuntu 14.04 LTS);[2], HDFS;[3], jupyterhub[4] MongoDB[5], Python 3;[6] and nginx[7]. All of this software is available also in previous revisions, from a variety of sources on the Internet.

The project prefers open source releases of its products. Open-source software generated by the project, including measurement tool as described in Section 4 as well as any code developed for the Observatory itself for management and analysis purposes, are made available and archived on GitHub, at `https://github.com/mami-project`. We commit to maintaining access to the software necessary for interacting with data in the Observatory as long as the Observatory is in operation.

Other software that is not directly related to the operation of the Observatory may not made generally available to the public and is stored in the project's github repository at `gitlab.mami-project.eu`, where it is backed up daily and are transferred to off-site storage once a week.

[1] `http://pandas.pydata.org`
[2] `http://releases.ubuntu.com/14.04/`
[3] `https://hadoop.apache.org/`
[4] `https://github.com/jupyter/jupyterhub`
[5] `https://www.mongodb.org/`
[6] `https://www.python.org/`
[7] `https://www.nginx.com/`

3 Data Management Plan

3.1 Data Set Description

The measurement work in the MAMI project is largely about assessing and classifying the types and extent of impairments to path transparency in the Internet by middleboxes. The data generated and stored by MAMI therefore consists of empirical observations of paths that exist in the Internet, and the conditions that exist along with them. A typical observation will measure whether on a certain date a certain condition was true on a certain path; for example, "On 1 April 2016, at 12:34:45 UTC, it was possible to establish a TCP connection from 2001:db8::dead:beef to 2001:db8:abcd::1 on port 443." Condition information is often derived from observation of packets and packet headers resulting from certain active Internet measurement activity; conditions may also contain additional data about measured parameters associated with the condition (e.g., round-trip time as measured by packets with given properties).

More formally, an observation consists of:

- one or more **timestamps** at which the observation is considered valid, either because it was directly measured or derived from a direct measurement taken at these times;
- a description of the **path** for which the observation is considered valid, consisting of a sequence of one or more path elements (i.e. addresses, prefixes, autonomous system numbers, or data-source-generated pseudonyms therefor);
- a description of the **condition** observed along the path, as defined by the analysis module generating the observation;
- any additional **values** associated with the condition; and
- a reference to the **source** of the observation, both the raw data (from which metadata is available) as well as the version of the analysis module that generated it.

Observations are either directly collected from vantage points around the Internet, or they are computed from raw measurement data. Data uploaded to MAMI comes from publicly available cloud servers as well as from Internet-connected testbeds. The measurement run may be simple active measurements such as pings or traceroutes, or they may be more elaborate exchanges of protocol messages.

Some data are generated explicitly for MAMI. For example, Pathspider (see Section 4.1) is a tool developed within the project. Pathspider does active A/B testing of connectivity dependency and feature usability of optional transport features (e.g., Explicit Congestion Notification (ECN), and the risk of enabling ECN by default on the client side). In order to determine the parameters of optional transport features, it is necessary to keep track of certain IP and TCP header fields, something that most data sets do not do. Pathspider will in turn be deployed on vantage points including hosts from the Measuring Mobile Broadband Networks in Europe (MONROE) project.

MAMI data are useful to all networking researchers interested in path-related issues. For MAMI itself, this is mainly path transparency, but it could also be about connectivity or even about

certain protocols. For example, MAMI data might be useful to determine connectivity to and from countries whose governments aim to control or monitor their citizens' use of the Internet.

MAMI has not yet generated any data, but MAMI-like data have in the past been used for scientific publications; for example, on using path transparency observations to support protocol engineering [7], on middlebox cooperation [9], or on the Internet-wide deployment of ECN [8].

3.2 Standards and Metadata

MAMI data consist of observations of path conditions, and raw data from which these observations can be derived. Raw data generated by the project or imported to the measurement data must contain at least the following metadata, derived from the metadata available from each data source.

- A (low-precision) timestamp at which the measurement data was created
- A (low-precision) timestamp at which the measurements were added to the Observatory
- Information about the entity (organization, individual, etc.) supplying the data
- Information about any licensing terms that may apply to the data
- Any URL for automated retrieval / re-retrieval of the data

Information about the source and target of active measurements, and timestamps for each part of the measurement, from which path and timestamp information in the observations are derived, are stored in the data itself.

Currently, raw data is stored as JSON [2], CSV [4], and IPFIX [1]; derived observations are stored as JSON.

3.3 Data Sharing

As described in Section 1, due to privacy concerns MAMI will not provide general access to raw data sets to external users. These raw data are stored inside MAMI with "copyright MAMI consortium, all rights reserved". MAMI can license these data for use by other researchers on a case-by-case basis, after these researchers have come to an agreement with the MAMI consortium to access the raw data and not expose any PII in any derived results.

The results from the MAMI query interface, which provides public access to the observations stored in the MAMI database, are licensed under the Creative Commons "Attribution 4.0 International" (CC BY 4.0) license (see `https://creativecommons.org/licenses/by/4.0/`).

The MAMI Observatory is open to all data sets where MAMI query results involving such data sets can be shared using CC BY 4.0. However, storing MAMI-generated data in the Observatory always has priority over third-party data, and third-party data may be removed from the Observatory should space become an issue. This is obviously not a problem for cached data sets, but MAMI will even try to find a home for data sets that are not merely cached and that would otherwise be orphaned, on a best-effort basis.

Data sets that are uploaded to MAMI, but which are later found not to be compatible with the MAMI data-sharing license, may be removed without notice.

3.4 Archiving and Preservation (Incl. Storage and Backup)

At the time of this first version of the DMP, the Observatory does not contain live data. Plans for archiving are thus preliminary and fluid.

Only data that are originally generated by MAMI will be archived and curated. At the time of writing, this includes Pathspider data and certain tracebox or copycat data sets.

For data set storage and backup, ZHAW will back up the HDFS and MongoDB onto external disk drives. Several drives will be used in rotation, and at least one drive will always be stored off-site.

After the end of the project, ZHAW will prepare a final, unalterable version of the data. These data will then be made available to researchers on request. Support for the MAMI web site and public-facing repository query tools will be continued as long as funds are available to sustain this long-term curation.

4 Data Sources

The MAMI Observatory is managed as a single, unitary data set, containing both observations for querying, as well as raw data for analysis and reanalysis from which these observations are derived. In this section, we list the data sources we presently know will provide data to the MAMI Observatory. This list is not complete, and will expand in the future. Whenever measurement from a new data source is added to the Observatory, it is further necessary to add new analysis modules to generate observations from this data.

4.1 PathSpider

PathSpider is a generalized tool for building connectivity and optional transport feature / transport protocol A/B functionality tests. A/B testing differentiates transient connectivity failures from connectivity failures due to the use of a particular transport protocol or feature. It currently supports testing of ECN connectivity and negotiation, but work is presently underway to add support for Multipath TCP, TCP Fast Open, TCP window scale negotiation, Stream Control Transmission Protocol (SCTP), and other protocols and protocol features.

PathSpider functions by generating two simultaneous flows from the source to the target, and passively observing these flows at the source to determine the characteristics of the flow. The raw data generated by this observation process are essentially flow data, linking characteristics of the “A” flow (feature enabled) to those of the “B” flow (feature disabled, experimental control). These raw flow data are analyzed by the PathSpider tool itself into MAMI-native observation records before transmission to the Observatory.

PathSpider’s output therefore consists of $\{time, path, condition, value\}$ tuples as in section 3.1:

timestamps Observation time of the first packet in the flow from which the condition was derived.

path Path derived from the source and destination addresses of the measurement.

condition Condition observed along the path; for example “ECN negotiation successful”, “ECN negotiation causes connectivity failure”.

values An value associated with the condition (not yet required by present conditions, for future use).

source reference Version of Pathspider used (in terms of GitHub tag or commit hash).

Pathspider was designed for large-scale testing of millions of targets (e.g. the Alexa top million webservers) from a set of active measurement agents; past measurement campaigns have used DigitalOcean cloud server instances as active measurement agents. However, for future verification of path support for uncommon features (e.g. SCTP), it may be necessary to operate targets with known support for these features, and passive observation of the traffic at the targets can also be used to generate observations.

4.2 Tracebox

Tracebox experiments attempt to contact a remote server from a vantage point and identify the used path. Tracebox data sets are in a binary format called warts, native to CAIDA's Scamper tool[1] on which it is based. These data are translated into a Tracebox-specific JSON schema for analysis at the observatory.

Each data set contains necessary metadata about the measurement, as follows:

version The version of the tool used to obtain the data set.

type The type of data contained in the data set, e.g., `tracebox`.

userid The (Unix) user ID under which the tool ran. Often `0`.

method The method used to gather data, e.g., `ip4-tcp`.

probe The basic probe format, e.g., `ip/tcp/mss(1460)/sackp` (i.e., TCP segment, with Selective Acknowledgment and MSS of 1460 bytes).

src The vantage point's source address.

dst The experiment's destination address.

sport/dport The experiment's source and destination TCP ports (if TCP is being used).

result The overall result, e.g., `success`

start The experiment's start time, containing `sec` (seconds since the Epoch), `usec` (microseconds) and `ftime` (human-readable local time, e.g., `2016-03-23 11:57:10`).

Assorted TCP and IP options These flags and values are often informative only, such as `tcp_seq` (initial TCP sequence number), but may sometimes be important for the protocol, such as `tcp_ack`.

4.3 Copycat

Copycat is a tool for detecting differential treatment of UDP and TCP traffic over an Internet path between two measurement agents. Copycat generates raw IPFIX [1] flow data using the QoF [6] flow meter, which contains additional information about TCP loss and latency. By comparing the characteristics of UDP traffic with TCP traffic along the same path at equivalent times, differential treatment can be detected.

Raw Copycat data contains basic QoF flows as well as TCP performance metrics; i.e., the following IPFIX Information Elements (IEs) as well as the appropriate reverse counterparts [5].

- octetDeltaCount
- packetDeltaCount

[1] `https://www.caida.org/tools/measurement/scamper/`

- protocolIdentifier
- tcpControlBits
- sourceTransportPort
- sourceIPv4Address
- ingressInterface
- destinationTransportPort
- destinationIPv4Address
- egressInterface
- sourceIPv6Address
- destinationIPv6Address
- minimumTTL
- maximumTTL
- flowEndReason
- flowId
- flowStartMilliseconds
- flowEndMilliseconds
- transportOctetDeltaCount
- transportPacketDeltaCount
- initialTCPFlags (6871 / 14)
- unionTCPFlags (6871 / 15)
- reverseFlowDeltaMilliseconds (6871 / 21)
- reverseInitialTCPFlags (6871 / 16398)
- reverseUnionTCPFlags (6871 / 16399)
- tcpSequenceCount (35566 / 1024)
- tcpRetransmitCount (35566 / 1025)
- minTcpRttMilliseconds (35566 / 1029)
- ectMarkCount (35566 / 1031)
- ceMarkCount (35566 / 1032)
- tcpSequenceLossCount (35566 / 1035)

- tcpLossEventCount (35566 / 1038)
- qofTcpCharacteristics (35566 / 1039)
- tcpRttSampleCount (35566 / 1046)

See `https://github.com/britram/qof/wiki` and `https://iana.org/assignments/ipfix` for a full reference for relevant IE definitions.

4.4 Revelio

Revelio [3] is a tool for detecting IPv4 network address translation on access networks. Revelio produces CSV-formatted data with the following fields:

boxid Unique identifier of the device running the Revelio client (assigned based on MAC address).

revelio_type The Revelio version and the platform used to deploy Revelio.

timestamp Time at the start of the measurement.

local_IP IP address of the device running the Revelio client.

IGD IP address of the WAN-facing interface if device supports UPnP,

STUN_mapped The public mapped address (the GRA).

trace_packetSize The packet size of the traceroute probe.

traceroute_result Output of traceroute to a fixed target examining the hops within the access network

References

[1] Benoit Claise, Brian Trammell, and Paul Aitken. Specification of the IP flow information export (IPFIX) protocol for the exchange of IP traffic flow information. RFC 7011, Internet Engineering Task Force, September 2013.

[2] ECMA International. The JSON data interchange format. Standard ECMA-404, ECMA International, Rue du Rhône 114, CH-1204 Geneva, October 2013.

[3] Andra Lutu, Marcelo Bagnulo, Amogh Dhamdhere, and kc claffy. Nat revelio: Detecting nat444 in the ISP. In *Proceedings of the 2016 Passive and Active Measurement Conference*, Heraklion, Mar 2016.

[4] Yakov Shafranovich. Common format and MIME type for CSV files. RFC 4180, Internet Engineering Task Force, October 2005.

[5] B. Trammell and E. Boschi. Bidirectional Flow Export using IP Flow Information Export (IPFIX). RFC 5103, RFC Editor, January 2008.

[6] Brian Trammell, David Gugelmann, and Nevil Brownlee. Inline data integrity signals for passive measurement. In *Proceedings of the 6th International Workshop on Traffic Monitoring and Analysis (TMA 2014)*, London, UK, 2014.

[7] Brian Trammell and Mirja Kühlewind. Observing Internet path transparency to support protocol engineering. In *Proceedings of the first IRTF/ISOC Workshop on Research and Applications of Internet Measurements (RAIM)*, Yokohama, Japan, Oct 2015.

[8] Brian Trammell, Mirja Kühlewind, Damiano Boppart, Iain Learmonth, Gorry Fairhurst, and Richard Scheffenegger. Enabling Internet-wide deployment of explicit congestion notification. In *Proceedings of the 2015 Passive and Active Measurement Conference*, New York, Mar 2015.

[9] Brian Trammell, Mirja Kühlewind, Elio Gubser, and Joe Hildebrand. A new transport encapsulation for middlebox cooperation. In *Proceedings of the 2015 IEEE Conference on Standards for Communications and Networking*, Tokyo, Japan, Oct 2015.